The multibus algorithm

Jean-Yves Le Boudec

IRIA

CENTRE DE RENNES

IRISA

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
BP 105
78153 Le Chesnay Cedex
France

Tél. (1) 39 63 55 11

Rapports de Recherche

N° 539

THE MULTIBUS ALGORITHM

Jean-Yves LE BOUDEC

Juin 1986

IRISA

INSTITUT DE RECHERCHE EN INFORMATIQUE
ET SYSTÈMES ALÉATOIRES

Campus Universitaire de Beaulieu
Avenue du Général Leclerc
35042 - RENNES CÉDEX
FRANCE
Tél. : (99) 36.20.00
Télex : UNIRISA 95 0473 F

THE MULTIBUS ALGORITHM

Jean-Yves LE BOUDEC
IRISA/INSA

Publication Interne n° 299
Juin 86
42 pages

ABSTRACT

Considered are product form queueing networks, in which some of the stations are of the MSCCC type (stations with multiple servers, in which two customers of the same group of classes cannot be served simultaneously). These networks can model parallel architectures with multiple buses. Usual product form network algorithms do not apply to such networks, due to the complexity of the function associated to the MSCCC station. The MULTIBUS algorithm presented here yields a mean value analysis of such closed networks. An algorithm for the open case is also given.

RESUME

On considère des réseaux de files d'attente à forme produit, dans lesquels certaines stations sont du type MSCCC (stations possédant plusieurs serveurs, dans lesquelles deux clients du même groupe de classes ne peuvent être servis simultanément). Ces réseaux modélisent, entre autres, des architectures parallèles à bus multiples. Les algorithmes standards pour réseaux à forme produit ne s'appliquent pas ici, à cause de la complexité de la fonction associée à une fonction MSCCC. L'algorithme MULTIBUS, présenté dans ce rapport, fournit une analyse par valeurs moyennes de tels réseaux fermés. Le cas ouvert est aussi évoqué.

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE
(L.A. 227)
UNIVERSITÉ DE RENNES 1 I.N.S.A. DE RENNES

INSTITUT NATIONAL DE RECHERCHE
EN INFORMATIQUE ET EN AUTOMATIQUE
(LABORATOIRE DE RENNES)

CONTENTS

1 - INTRODUCTION

Modelling multiple bus multiprocessor systems has been the object of many studies. K.B. Irani and I.H. Önyuksel obtained in [I-Ö 84] a closed form solution for a symmetrical markovian queueing model. Similarly, Marsan, Balbo, Chiola and Donatelli develop in [MBCD 84] generalized stochastic Petri nets models of multiple bus access, for which local balance hold.

More recently, an extension of the BCMP theorem has been presented in [Leb 85] : a new type of station (the "MSCCC" station) is introduced, that can model parallel access to shared resources, under exponential holding time and FIFO disciple assumptions. It is proven that product form is maintained when such stations are inserted in BCMP networks.

Nevertheless, usual product-form network algorithms do not directly apply to such networks ; this is due to the complexity of the function associated to MSCCC stations.

In this paper, we present the MULTIBUS algorithm, which yields mean values for such closed or open networks, thus avoiding the computation of the associate functions of the MSCCC stations.

MULTIBUS provides an exact, efficient analytical tool, which is able to model architectures with one or several multiple bus systems, such as partial bus multiprocessors. As is pointed out in [CMB 86], it can also apply to many situations where access to a shared resource is limited to a fixed number of users, with FIFO queueing discipline.

A brief description of the MSCCC station is recalled in section 2, and a new aggregation formula is presented in section 3. Section 4 is devoted to the MULTIBUS algorithm for closed networks (multiple chain or single chain case). The version for open networks is presented in section 5.

The paper is followed by an appendix, in which miscellaneous remarks about local balance, aggregation and equivalent service rates are given.

The following table lists most of the notation used in the paper. The meaning of every symbol that is not explicitly defined in the sequel should be found there.

Notation

$k=(m,r)$: a class of customers at a MSCCC station ; m is the group number, r is the chain number.

K : the population size (single chain case)

$\underline{K}=(K_1,\ldots,K_R)$: the population vector (multiple chain case : K_r is the population size of chain r).

θ_r : the mean number of visits to a given station by chain r customer.

$\theta_{m,r}=\theta_r q_{m,r}$: the mean number of visits to a given MSCCC station by class (m,r) customers.

$q_{m,r}=q^{\ell}_{m,r}$: the probability that an arriving chain r customer joins class (m,r) at a MSCCC station (number ℓ).

μ : the service rate at a MSCCC station.

$\rho_k,\rho_{m,r}=\theta_{m,r}/\mu$ (at a MSCCC station).

$\underline{x}=(x_{m,r})$: an aggregate state of a MSCCC station ; $x_{m,r}$ is the number of class (m,r) customers.

$\underline{\rho}^{\underline{x}}=\prod_{m,r}\rho_{m,r}^{x_{m,r}}$

$\underline{n}=(n_m)$: n_m is the number of group m customers : $n_m=\sum_r x_{m,r}$.

$a\wedge b$: the minimum of the two values a,b .

$nz(\underline{n})$: the number of non zero entries in $\underline{n}$.

$\eta(\underline{x})=\prod_m\left[n_m!/(\prod_r x_{m,r}!)\right]$

$\phi(\underline{n})$ = if $nz(\underline{n})<B$ then 1 else $(\sum_m \phi(\underline{n}-\underline{e}_m))/B$

$\underline{e}_m$ = (0,0,1,0,...0) where 1 is in the m^{th} position.

$\underline{1}_{m,r}$: defined by $\underline{1}_{m,r}(i,j)=0$ except $\underline{1}_{m,r}(m,r)=1$.

$\mu^*_k(\underline{y})$: the equivalent service rate for class k, given the state $\underline{y}$ of the station.

$\mathcal{N}^{[\ell]}$: the complement of station ℓ in the network $\mathcal{N}$.

$G(\underline{K}),G^{[\ell]}(\underline{K})$: the normalizing constant for $\mathcal{N}$ [resp. $\mathcal{N}^{[\ell]}$] when the population vector is $\underline{K}$.

$P_\ell(\underline{y}_\ell \mid \underline{K})$: the marginal steady-state probability of $\underline{y}_\ell$, (station ℓ), given the population vector is $\underline{K}$.

$K(\underline{y}_\ell)$: the population vector in station ℓ, state $\underline{y}_\ell$; for the MSCCC stations, $K(\underline{x}) \equiv \underline{k}$, with $k_r = \sum_m x_{m,r}$.

$\lambda_r(\underline{K}), \lambda(K)$: the throughput at a given station (for chain r)

$\lambda_r^{[\ell]}(\underline{K}), \lambda^{[\ell]}(K)$: the throughput in the complement network (for chain r).

2 - THE MSCCC STATION

2-1 The station

Definition of the MSCCC station

We recall the definition of the MSCCC station ([LeB 85]) :

* There are B identical exponential servers (with service rate μ) and M groups of classes of customers.

* Upon arrival, a customer enters the service center if at least one server is available and if there is no customer of the same group already receiving service ; otherwise the customer joins the waiting-room.

In the sequel, we shall say that a group is active if one customer of the group is receiving service.

* Customers in the waiting-room are ordered according to their arrival epochs ; whenever one or more server becomes available (due to a departure from the station), the first customer belonging to a non-active group enters the service station.

In other words, the service discipline is on a first come first served basis, but two customers of the same group cannot be served simultaneously. In particular, a customer can be kept waiting either because no server is available, or because its group is already active.

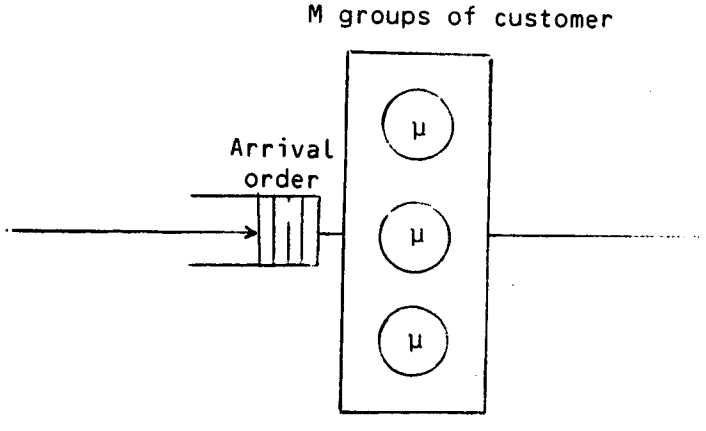


Figure 1 : The MSCCC station

In the MSCCC station, a class of customers is noted (m,r), where m is the number of the group it belongs to ; r is the chain number of the class. In the single chain case; a class is simply noted m.

Example : a partial bus multiprocessor system

Consider a system in which P processors access M × G memory modules through G identical B-uple buses. Each multiple bus gives access to a set of M memory modules. (Fig. 2). The m^{th} memory module of the g^{th} group is noted (g,m). It is assumed that a request issued by the p^{th} processor is directed to memory module (g,m) according to a Bernoulli trial, with probability $q_{p,g,m}$.

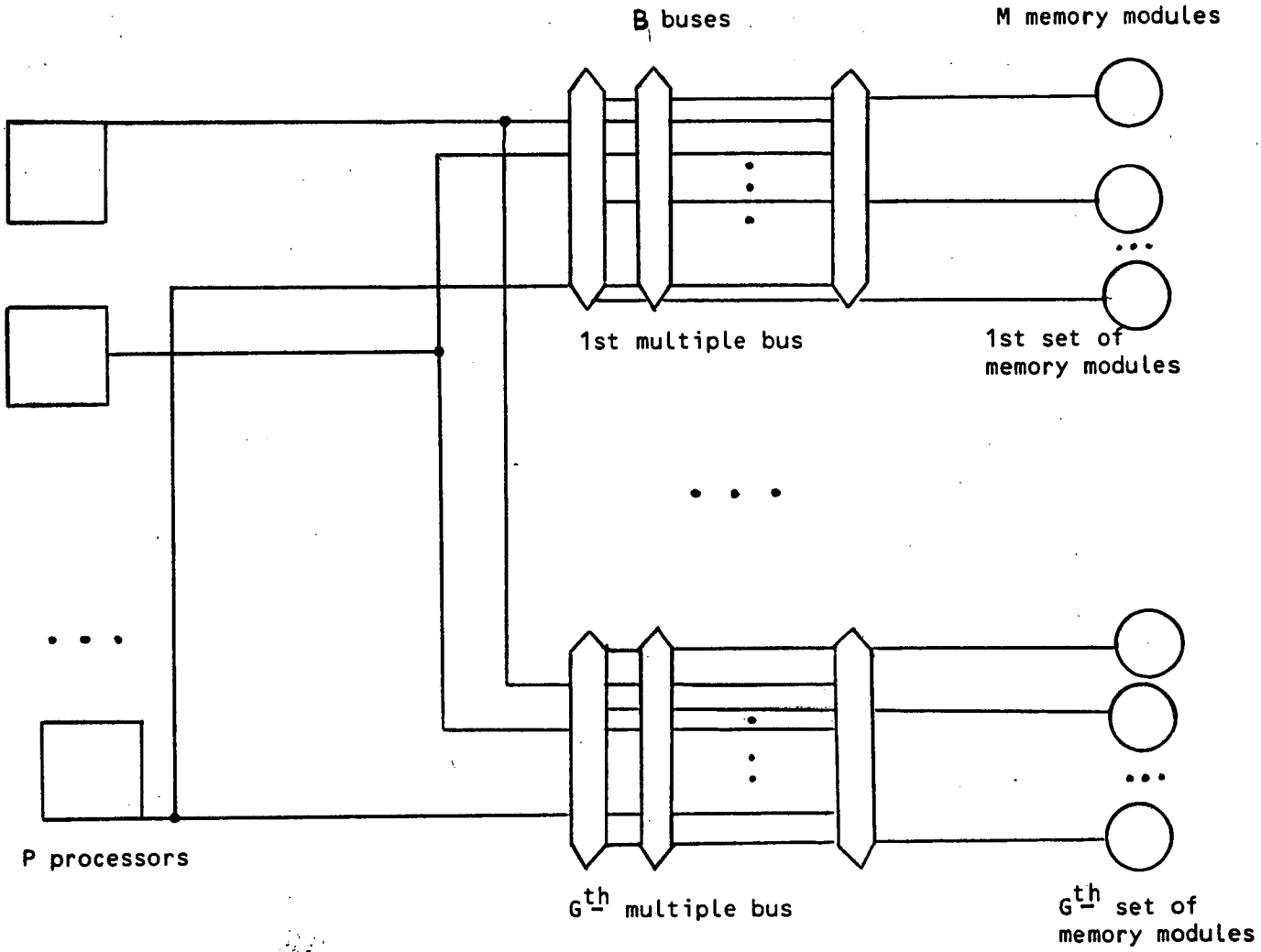


Figure 2 : A partial bus multiprocessor architecture

Such a system can be modelled by a network consisting of a P-server station (the PROCESSOR station) and G MSCCC stations (the BUS stations) with B servers and M groups of class. There are P chains of customers (one per processor), each chain containing one customer. On leaving the PROCESSOR station, a class p customer joins the BUS(g) station with probability $\sum_m q_{p,g,m}$, and enters class (m,p) at the station with probability $q_{p,g,m} / \sum_m q_{p,g,m}$ (Fig. 3). The amount of time spent in a processor may have any coxian distribution, with mean $1/\lambda$; it is assumed that the total amount of time spent in accessing a memory module is exponentially distributed, with mean $1/\mu$. By an appropriate choice of $q_{p,g,m}$, it is possible to allow each processor to have one or more favorite memory modules, which are more often visited.

On the contrary, if $q_{p,g,m}$ does not depend on processor p, then a single chain model with a total population of P customers is preferable.

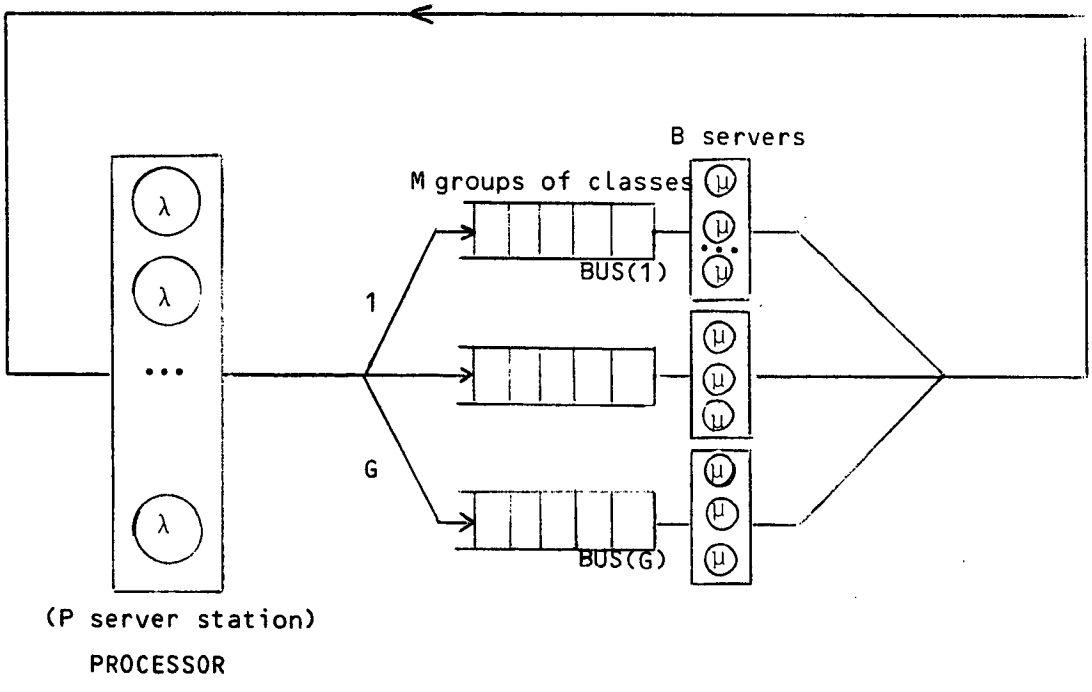


Figure 3 : Modelling a partial bus multiprocessor architecture by a product-form network

2-2 The steady-state probability

Consider a network of L stations that satisfies the hypotheses of the

BCMP theorem, except that some stations may be of the MSCCC type. Then for every station ℓ there exists a function f_ℓ such that the steady-state probability of the network is given by :

$$(2\text{-}1) \quad P(\underline{y}_1,\ldots,\underline{y}_L) = G(\underline{K}) \prod_{\ell=1}^{L} f_\ell(\underline{y}_\ell)$$

where y_ℓ is an appropriate description of the state of station ℓ, and $G(\underline{K})$ is a normalizing constant, depending on the total population vector $\underline{K}$.

In the sequel, function f_ℓ is called the function associated to station ℓ. For the MSCCC station, a state $\underline{y}_\ell$ is $(AC,\underline{c})$ where AC is the set of all active classes (classes of customers receiving service), and $\underline{c} = (c_1,c_2,\ldots,c_S)$ is the list of the waiting customers, in arrival order. (Subscript ℓ is omitted). The associate function f_ℓ is then given by

$$(2\text{-}2) \quad f_\ell(AC,\underline{c}) = \prod_{\gamma \in AC} \rho_\gamma \prod_{i=1}^{S} (\rho_{c_i} N_{A,\underline{c}}(i))$$

where $N_{A,\underline{c}}(i)$ is defined as follows :

- A is the set of all active groups of classes
- if there are only non active class customers waiting in positions 1 through i of the waiting room, then $N_{A,\underline{c}}(i)$ is the number of those active groups of classes (i.e., the number of active groups of classes that are represented in positions 1 to i of the waiting-room).
- if there is (at least) one customer of non active class in positions 1 through i, then $N_{A,\underline{c}}(i) = B$ (the number of servers).

(See [LeB 85] for more details).

In the open case, the stability condition is :

$$(2\text{-}3) \quad \begin{cases} \sum_{(m,r)} \rho_{(m,r)} < B \\ \forall\, m,\ \sum_{r} \rho_{(m,r)} < 1 \end{cases}$$

It expresses that, on one side, the total intensity factor is less than the number of servers, and on the other side, that the intensity factor of every group of classes is less than 1.

Note that it is possible to allow the service rate μ to depend on the total number of customers in the station.

3 - AGGREGATION FORMULAE

3-1 The aggregate MSCCC station

Formula (2-2) refers to much detailed states, which, together with the complexity of the computation of $N_{A,\underline{c}}(i)$, makes it untractable in practice. It is thus necessary to obtain the steady-state probability of aggregate states. A first result is given in [LeB 85], which helped prove the stability condition (2-3). In this section we give a simpler aggregation formula, which will be used in the algorithms of section 4.

A state of the aggregate MSCCC station is defined to be $\underline{x}$, where $x_{m,r}$ is the number of class (m,r) customers in the station (including those in the service center). In the sequel, the phrase "micro-state" refers to $(AC,\underline{c})$. Also define :

- $n_m = \sum_r x_{m,r}$ (the number of group m customers).
- $nz(\underline{n}) = \sum_m 1_{(n_m>0)}$ (the number of non-zero entries in $\underline{n}$)
- $\eta(\underline{x}) = \prod_m \left[n_m ! \,/\, \pi_r \, (x_{m,r}!) \right]$

Proposition 1 (aggregate steady-state probability for the MSCCC station)

In a product-form queueing network as considered in section 2-1, the function associated to an aggregate MSCCC station is given by :

(3-1) $\quad f(\underline{x}) = \phi(\underline{n})\ \eta(\underline{x})\ \underline{\rho}^{\underline{x}}$

where ϕ is defined by

(3-2) $$\begin{cases} \phi(\underline{o}) = 1 \\ (nz(\underline{n}) \wedge B)\ \phi(\underline{n}) = \sum_m \phi(\underline{n} - \underline{e}_m) \end{cases}$$

with the convention that $\phi(\underline{n}) = 0$ if some entry of $\underline{n}$ is negative.

(3-3) Besides, if $nz(\underline{n}) \leq B$ then $\phi(\underline{n}) = 1$

Note that the factor η is due to the existence of several chains of customers, wheras ϕ is typical of the MSCCC station. Also note that ϕ is iden-

tical to the weight function W defined in [I-Ö].

Proof : The function associated to a station in a product form network is equal up to a multiplicative constant, to the steady-state probability of the queue in isolation. Besides, formula (2-1) still holds at the aggregate level, with the same normalizing constant, provided some compatibility condition is satisfied. (See appendix for more detail). Roughly speaking, this condition means that no infeasible state is introduced in the aggregation process.

It is thus sufficient to show the proposition when the network is reduced to a MSCCC station in isolation, i.e., fed in by independant Poisson processes, with parameter $\theta_{m,r}$ for class (m,r). It is now sufficient to consider the case where R = 1 ; thus $\underline{n} = \underline{x}$, and $\theta_{m,r}$ is simply noted θ_m .

For every micro-state e of the MSCCC station, due to local balance, the following equation holds :

$$(3\text{-}4) \quad P(e)\mu(e) = \sum_m \theta_m \sum_{e' \in \mathscr{E}_m(e)} P(e')$$

where P is the steady state probability (for micro-states), $\mu(e)$ is the output rate from state e, and $\mathscr{E}_m(e)$ is the set of all states from which a class m arrival results in state e. ($\mathscr{E}_m(e)$ has 0 or 1 element).

Note that if micro-state e aggregates to $\underline{n}$, then $\mu(e) = [B \wedge nz(\underline{n})]\mu$, hence :

$$(3\text{-}5) \quad P(\underline{n})(nz(\underline{n}) \wedge B)\mu = \sum_m \theta_m \sum_{\ell' \in \mathscr{E}_m(e),\, e \in \underline{n}} P(e')$$

("$e \in \underline{n}$" is short for "micro-state e aggregates to $\underline{n}$"; besides , the same notation is used for the steady-state probability of micro and aggregate states).

Now every micro-state e' in $\underline{n} - \underline{e}_m$ is element of $\mathscr{E}_m(e)$ for one and only one micro-state e', hence :

$$P(\underline{n})[nz(\underline{n}) \wedge B]\mu = \sum_m \theta_m P(\underline{n} - \underline{e}_m)$$

which proves (3-2).

Now (3-2) is clearly satisfied by any constant, as long as $nz(\underline{n}) \leq B$, which proves (3-3).

Examples

. If B = M, then $\phi(\underline{n})$ reduces to 1 ; in that case, the MSCCC station is indeed equivalent to M separate aggregate FIFO stations, for which formula (3-1) is the expression of product form.

. If B = 1, then recursion (3-2) is the recursion of the FIFO station ; it results in

$$\phi(\underline{n}) = (n_1 + \ldots + n_M)!/(n_1! \ldots n_M!)$$

and formula (3-1) gives :

$$f(\underline{x}) = \Big(\sum_{m,r} x_{m,r}\Big)!/\Big(\prod_{m,r} (x_{m,r}!)\Big)\ \underline{\rho}^{\underline{x}}$$

as expected.

. In the general case, we have not been able to find a closed form expression for ϕ ; some values of ϕ are given in table 1.

3-2 The equivalent service rate

Another way of viewing the aggregation process of the preceeding section is to consider that the MSCCC station can be replaced by an equivalent multiclass markovian server, whose service rate for each class of customer depends on the load of the queue. For classical BCMP stations, the equivalent service rate for class k has the form :

(3-6) $\dfrac{\mu(N)}{W_k}\dfrac{N_k}{N}$, where N (resp. N_k) is the number of customers (resp. class k customers) in the station. In general, for a quasi-reversible queue, the equivalent service rate for class k is deduced from the associate function f by :

(3-7) $\mu_k^*(\underline{y}) = f(\underline{y} - \underline{1}_k)/f(\underline{y}) \times \theta_k$

where $\underline{y}$ is the aggregate state, and $\underline{y} - \underline{1}_k$ the aggregate state with one less class k customer. ([LR 79],[RL 80]). Applying this formula to the MSCCC station yields the equivalent service rate for class (m,r) :

The equivalent service rate for class (m,r)

(3-8) $\mu^*_{m,r}(\underline{x}) = \mu\, \sigma_m(\underline{n})\, x_{m,r}/n_m$

(3-9) where $\sigma_m(\underline{n}) = \phi(\underline{n} - \underline{e}_m)/\phi(\underline{n})$

Note that σ_m is the equivalent service rate for class m, in the single chain case, with unit service time : it is typical of the MSCCC station. Some properties of σ_m are listed below ; their proof is easy and is not given.

Properties of σ_m

(i) reduction to m=1 : $\sigma_m(\underline{n}) = \sigma_1(\underline{n}')$, where $\underline{n}'$ is the result of the permutation of n_1 and n_m

(ii) symmetry : $\sigma_1(\underline{n}) = \sigma_1(\underline{n}^*)$ where $\underline{n}^*$ is obtained from n by reordering $(n_2,\ldots,n_M)$ in decreasing order

(iii) total service rate $\sum_{m=1}^{M} \sigma_m(\underline{n}) = nz(\underline{n}) \wedge B$

hence $\sigma_1(n,n,n,\ldots,n,o,\ldots,o) = \left[nz(\underline{n}) \wedge B\right] / M$

Some values of $\sigma_1(\underline{n})$ are given in table 1. It can easily be seen that $\sigma_m(\underline{n})$ does not possess the general form for BCMP stations described in (3-6).

Viewing the MSCCC as a queue dependant markovian server is a possible way to obtain algorithms to solve product-form queueing networks with one or several MSCCC stations, using standard algorithms for this type of stations. This requires that σ_m be available for the algorithm. This is the point of view adopted by Chiola, Marsan and Balbo in [CMB 86], who develop algorithms that yield ϕ or σ_1 (or equivalent quantities). An alternative is to avoid the use of ϕ and σ_1 ; this is possible thanks to the basic recursions of section 4-2, which concern mean values of the network. They are the kernel of the MULTIBUS algorithm.

(n1,...,n5)	phi(n1,...,n5)
0 0 0 0 0	1
1 0 0 0 0	1
1 1 0 0 0	1
1 1 1 0 0	1
1 1 1 1 0	1.3333333
1 1 1 1 1	2.2222222
2 0 0 0 0	1
2 1 0 0 0	1
2 1 1 0 0	1
2 1 1 1 0	1.4444444
2 1 1 1 1	2.6666667
2 2 0 0 0	1
2 2 1 0 0	1
2 2 1 1 0	1.6296296
2 2 1 1 1	3.4074074
2 2 2 0 0	1
2 2 2 1 0	1.962963
2 2 2 1 1	4.7160494
2 2 2 2 0	2.6172839
2 2 2 2 1	7.1604938
2 2 2 2 2	11.934156
3 0 0 0 0	1
3 1 0 0 0	1
3 1 1 0 0	1
3 1 1 1 0	1.4814815
3 1 1 1 1	2.8641975
3 2 0 0 0	1
3 2 1 0 0	1
3 2 1 1 0	1.7037037
3 2 1 1 1	3.7942387
3 2 2 0 0	1
3 2 2 1 0	2.1234568
3 2 2 1 1	5.5171468
3 2 2 2 0	2.9958848
3 2 2 2 1	8.9026063
3 2 2 2 2	15.848194
3 3 0 0 0	1
3 3 1 0 0	1
3 3 1 1 0	1.8024691
3 3 1 1 1	4.3319616
3 3 2 0 0	1
3 3 2 1 0	2.3497942
3 3 2 1 1	6.6886145
3 3 2 2 0	3.563786
3 3 2 2 1	11.582076
3 3 2 2 2	22.147538
3 3 3 0 0	1
3 3 3 1 0	2.6831276
3 3 3 1 1	8.477366
3 3 3 2 0	4.4581618
3 3 3 2 1	15.893918
3 3 3 2 2	32.743483
3 3 3 3 0	5.9442158
3 3 3 3 1	23.173296
3 3 3 3 2	51.382409
3 3 3 3 3	85.637346

(n1,...,n5)	sigma
0 0 0 0 0	0
1 0 0 0 0	1
1 1 0 0 0	1
1 1 1 0 0	1
1 1 1 1 0	0.75
1 1 1 1 1	0.6
2 0 0 0 0	1
2 1 0 0 0	1
2 1 1 0 0	1
2 1 1 1 0	0.92307692
2 1 1 1 1	0.83333333
2 2 0 0 0	1
2 2 1 0 0	1
2 2 1 1 0	0.88636363
2 2 1 1 1	0.7826087
2 2 2 0 0	1
2 2 2 1 0	0.83018868
2 2 2 1 1	0.72251309
2 2 2 2 0	0.75
2 2 2 2 1	0.65862069
2 2 2 2 2	0.6
3 0 0 0 0	1
3 1 0 0 0	1
3 1 1 0 0	1
3 1 1 1 0	0.975
3 1 1 1 1	0.93103448
3 2 0 0 0	1
3 2 1 0 0	1
3 2 1 1 0	0.95652174
3 2 1 1 1	0.89804772
3 2 2 0 0	1
3 2 2 1 0	0.92441861
3 2 2 1 1	0.8547986
3 2 2 2 0	0.87362637
3 2 2 2 1	0.80431433
3 2 2 2 2	0.75302943
3 3 0 0 0	1
3 3 1 0 0	1
3 3 1 1 0	0.94520547
3 3 1 1 1	0.8758708
3 3 2 0 0	1
3 3 2 1 0	0.90367776
3 3 2 1 1	0.82485644
3 3 2 2 0	0.84064664
3 3 2 2 1	0.76865377
3 3 2 2 2	0.71557361
3 3 3 0 0	1
3 3 3 1 0	0.87576687
3 3 3 1 1	0.78899677
3 3 3 2 0	0.79938462
3 3 3 2 1	0.72871116
3 3 3 2 2	0.6763953
3 3 3 3 0	0.75
3 3 3 3 1	0.68587214
3 3 3 3 2	0.63725083
3 3 3 3 3	0.6

Table 1
(B=3,M=5)

4 - THE MULTIBUS ALGORITHM FOR CLOSED NETWORKS WITH ONE OR SEVERAL MSCCC STATIONS

4-1 General setting ; presentation of the MULTIBUS algorithm : the networks to which the algorithms apply

Considered is a closed network of queues that satisfy the hypothesis of the BCMP theorem, except for one thing : some of the queues are MSCCC stations as defined in section 2. There are R closed chains of customers :

A customer of chain r proceeds in the network from one station to another, according to an irreducible Markov chain, with transition probabilities $Q^r_{\ell,\ell'}$. There are no changes of chain, and a customer of chain r can belong to class (m,r) at a MSCCC station with probability $q^{\ell}_{m,r}$ (where ℓ is the number of the station). In particular, group m can include customers belonging to different chains, and one chain of customers can feed several groups of classes at a given MSCCC station.

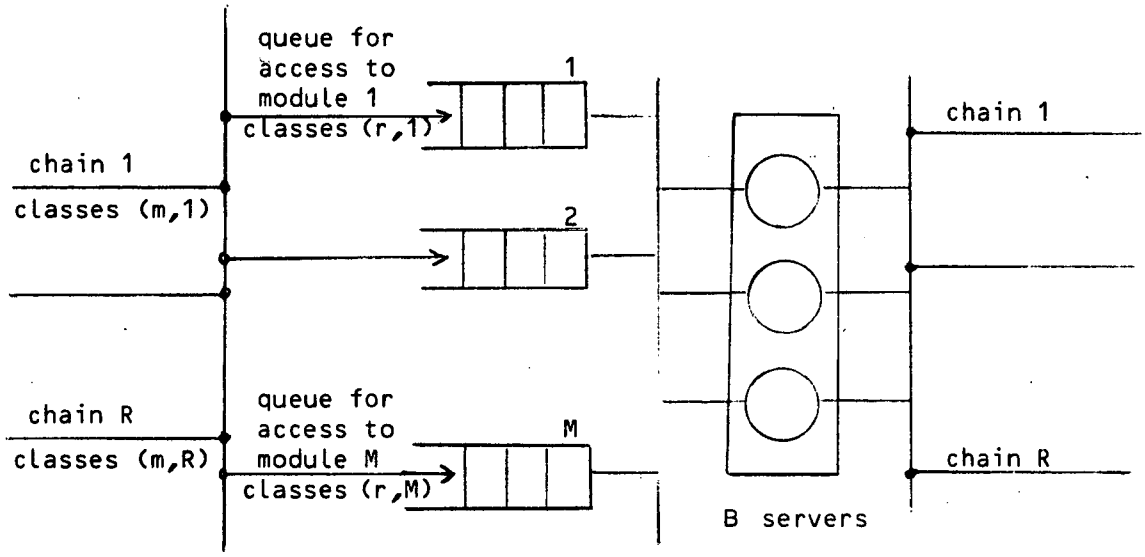


Figure 4 : MSCCC station in a network

The multiple chain case

The analysis of the network uses a decomposition procedure. Note $\mathcal{N}^{[\ell]}$ the "complement" of station ℓ in network $\mathcal{N}$: in $\mathcal{N}^{[\ell]}$, customers arriving at

station ℓ instantaneously proceed to another queue according to the routing policy previously defined. Then the marginal steady-state probability of station ℓ is related to $\mathcal{N}^{[\ell]}$ by the following formula ([R-L 80]) :

$$(4\text{-}1) \quad P_\ell(\underline{y}_\ell \mid \underline{K}) = f_\ell(\underline{y}_\ell)\, G^{[\ell]}(\underline{K} - \kappa(\underline{y}_\ell))/G(\underline{K})$$

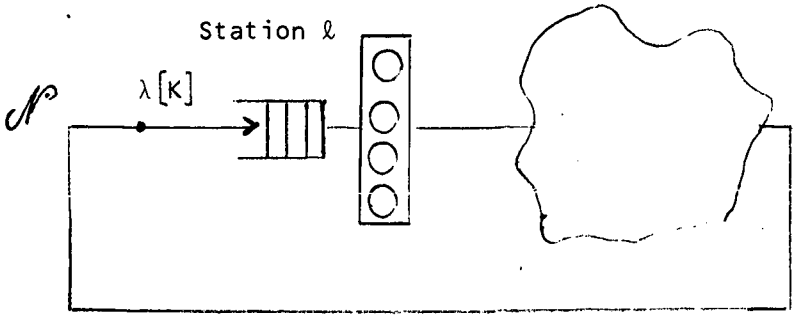
The MULTIBUS algorithm gets as input the array of normalizing constants for network $\mathcal{N}^{[\ell]}(K')$, $K' \leqslant K$. It produces the array of normalizing constant $G^{[\ell]}(\underline{K}')$ $\underline{K}' \leqslant \underline{K}$, together with some mean values for station ℓ in network $\mathcal{N}(\underline{K})$: mean queue length, distribution of the number of busy servers, utilisation of the servers and throughputs. The algorithm does not compute the steady-state distribution of the population of the station, but requires the storage of normalizing constants.

In order to analyze a network with one or more MSCCC stations, one should first analyse the complement of the MSCCC stations by means of classical product-form algorithms, then use the MULTIBUS algorithm to analyze the complete network.

The single chain case

In the single chain case, the MULTIBUS algorithm is able to produce the throughput $\lambda(\underline{K})$ without computing the normalizing constant. The single chain MULTIBUS algorithm gets as input the array of throughputs $\lambda^{[\ell]}(K')$, $K' \leqslant K$, for the complement network, and produces the array $\lambda(K')$ of throughputs for station ℓ in network $\mathcal{N}$, together with the same mean values as in the multiple chain case.

It is known [CHW], [COU], [REI], that $\mathcal{N}^{[\ell]}$ can be replaced by a queue dependant server, the rate of which is equal to $\lambda(K')$:



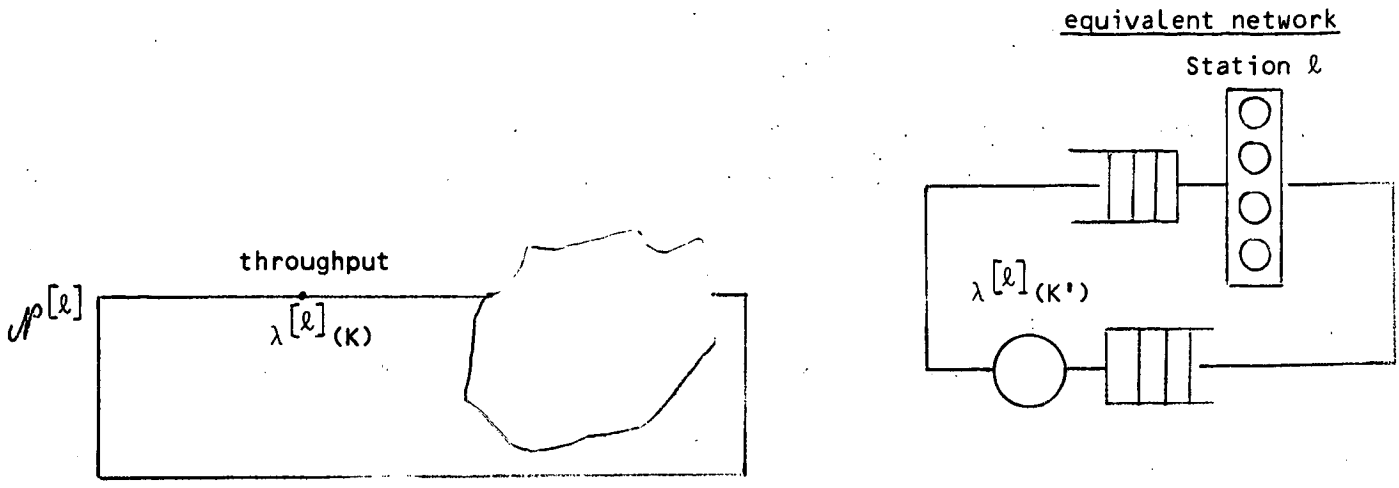


Figure 5

Thus, an alternative definition of the single chain MULTIBUS algorithm is that it solves the loop consisting of a MSCCC station and a queue dependant server.

4-2 The basic recursions

Classical algorithms for product form networks use some recursions involving the network (over population or stations). In the special case of the MSCCC station it is necessary to introduce some recursions concerning the station alone, so as to avoid the use of function ϕ defined in (3-2), or of the equivalent service rate.

The following notation is local to this section :

$$SN_b(\underline{k}) = \{\underline{x}/\kappa(\underline{x}) = \underline{k},\ nz(\underline{n}) = b\}$$

$$SR_{b,m}(\underline{k}) = \{\underline{x} \in SN_b(\underline{k}),\ n_m > 0 \text{ and } n_{m+1} = \ldots = n_M = 0\}$$

$$SQ_{b,m}(\underline{k}) = \{\underline{x} \in SN_b(\underline{k}),\ n_m = n_{m+1} = \ldots = n_M = 0\}$$

$$P1(\underline{k}) = \sum_{\underline{x}/\kappa(\underline{x})=\underline{k}} f(\underline{x})$$

$$NB1_r(\underline{k}) = \sum_{\underline{x}/\kappa(\underline{x})=\underline{k}} f(\underline{x})\, x_{M,r}$$

$$N1_{b,r}(\underline{k}) = \sum_{\underline{x} \in SN_b(\underline{k})} f(\underline{x})\, x_{M,r}\ , \qquad NA1_b(\underline{k}) = \sum_{r=1}^{R} N1_{b,r}(\underline{k})$$

$$R1_{b,m}(\underline{k}) = \sum_{\underline{x} \in SR_{b,m}(\underline{k})} f(\underline{x})$$

$$Q1_{b,m}(\underline{k}) = \sum_{\underline{x} \in SQ_{b,m}(\underline{k})} f(\underline{x})$$

$NA1_b(\underline{k})$, $NB1_r(\underline{k})$, $N1_{b,r}(\underline{k})$ and $Q1_{b,M+1}$ are the quantities of interest, which after normalization, yield the mean queue length and bus utilization. The following proposition states the basic recursion :

Proposition 2

(i) for $1 \leq b \leq B$:

$$N1_{b,r}(\underline{k}) = \rho_{M,r}\, NA1_b(\underline{k} - \underline{e}_r) + \rho_{M,r}\,(R1_{b,M}(\underline{k}-\underline{e}_r) + Q1_{b-1,M}(\underline{k}-\underline{e}_r))$$

(ii) for $1 \leq b \leq B$ and $b \leq m \leq M$:

$$R1_{b,m}(\underline{k}) = \sum_{r=1}^{R} \rho_{m,r}(R1_{b,m}(\underline{k}-\underline{e}_r) + Q1_{b-1,m}(\underline{k}-\underline{e}_r))$$

(iii) for $1 \leq b \leq B$, $b+1 \leq m \leq M$:

$$Q1_{b,m}(\underline{k}) = Q1_{b,m-1}(\underline{k}) + R1_{b,m-1}(\underline{k})$$

(iv) $$\sum_{b=1}^{B} b\, N1_{b,r}(\underline{k}) = \sum_{r'=1}^{R} \left(\sum_{m} \rho_{m,r'}\right) NB1_r(\underline{k}-\underline{e}_{r'}) + \rho_{M,r}\, P1(\underline{k}-\underline{e}_r)$$

Proof :

- Formula (iii) is clear ; recall that :

$$f(\underline{x}) = \phi(\underline{n})\, \eta(\underline{x})\, \underline{\rho}^{\underline{x}}$$

and

$$\phi(\underline{n}) = 1 \quad \text{if} \quad nz(\underline{n}) \leq B$$

Now it can easily been seen from the explicit expression of η that :

$$\eta(\underline{x})\, x_{M,r} = \eta(\underline{x} - \underline{1}_{M,r})\, n_M$$

Hence, using (3-2) :

$$N1_{b,r}(\underline{k}) = \rho_{M,r} \sum_{\underline{x} \in SN_b(\underline{k})} f(\underline{x} - \underline{1}_{M,r})\, n_M$$

Changing variable $\underline{x} - \underline{1}_{M,r}$ to $\underline{x}$ yields :

$$N1_{b,r}(k) = \rho_{M,r}\ S1 + \rho_{M,r}\ S2$$

where

$$S1 = \sum_{\substack{\underline{x}/\kappa(\underline{x})=\underline{k}-\underline{e}_r \\ nz(\underline{n}+\underline{e}_M)=b}} f(\underline{x})\ n_M \quad , \quad S2 = \sum_{\substack{\underline{x}/\kappa(\underline{x})=\underline{k}-\underline{e}_r \\ nz(\underline{n}+\underline{e}_M)=b}} f(\underline{x})$$

First note that if $x_{M,r} > 0$ then $n_M > 0$ and thus $nz(\underline{n}+\underline{e}_M)=nz(\underline{n})$; hence :

$$S1 = \sum_{\underline{x} \in SN_b(\underline{k}-\underline{e}_r)} f(\underline{x})\, n_M = N1_{b,r}(\underline{k}-\underline{e}_r)$$

Then note that :

$$[nz(\underline{n}+\underline{e}_M)=b] \Longleftrightarrow [(nz(\underline{n})=b, n_M > 0) \quad \text{or} \quad (nz(\underline{n})=b-1,\ n_M = 0)]$$

hence

$$S2 = \sum_{\underline{x} \in SR_{b,M}(\underline{k}-\underline{e}_r)} f(\underline{x}) + \sum_{\underline{x} \in SQ_{b-1,M}(\underline{k}-\underline{e}_r)} f(\underline{x}) = R1_{b,M}(\underline{k}-\underline{e}_r) + Q1_{b-1,M}(\underline{k}-\underline{e}_r)$$

which proves (i). The proof of (ii) is quite similar, starting with : $\eta(\underline{x})1_{n_M>0} = \sum_r \eta(\underline{x}-\underline{1}_{M,r})$

- The proof of (iv) is a consequence of relation (3-5) (local balance) :

$$(4\text{-}2) \quad f(\underline{x})\ \mu\,[nz(\underline{n}) \wedge B] = \sum_{m,r'} f(\underline{x}-\underline{1}_{m,r'})\ \theta_{r'}\, q_{m,r'},$$

multiplying (4-2) by $x_{m,r}$ and summing over all $\underline{x}$ such that $\kappa(\underline{x})=\underline{k}$ yields :

$$\mu \sum_{b=1}^{B} b\ N1_{b,r}(\underline{k}) = \Big[\sum_{m,r'} \theta_{r'}\, q_{m,r'}, NB1_r(\underline{k}-\underline{e}_{r'})\Big] + \theta_r\ q_{M,r}\ P1(\underline{k}-\underline{e}_r)\ .$$

4-3 The MULTIBUS algorithm, multiple chain case

The following notation is used in the section :

let : $\chi_{M,r}$ denote the number of class (M,r) customers at time t_o ,

β the number of busy servers at time t_o ,

ν_m the number of group m customers at time t_o ,

and let : $NB_r(\underline{K}) \triangleq \mathbb{E}(\chi_{M,r})$

$$N_{b,r}(\underline{K}) \triangleq \mathbb{E}(\chi_{M,r}\ \mathbb{1}_{(\beta=b)}) \quad , \quad NA_b(\underline{K}) = \sum_r N_{b,r}(\underline{K})$$

$$R_{b,m}(\underline{K}) \overset{d}{=} \mathbb{P}(\nu_m > 0,\ \nu_{m+1} = \ldots = \nu_M = 0,\ \mathbb{1}_{(\beta=b)})$$

$$Q_{b,m}(\underline{K}) \overset{d}{=} \mathbb{P}(\nu_m = \ldots = \nu_M = 0,\ \mathbb{1}_{(\beta=b)})$$

$$P_b(\underline{K}) = \mathbb{P}(\beta=b) = (Q_{b,M}+R_{b,M})(\underline{K}) = Q_{b,M+1}(\underline{K})$$

where $\mathbb{E}$ (resp. $\mathbb{P}$) denotes the steady-state expectation (resp. probability), given the total population vector is $\underline{K}$.

Thus P_b gives the distribution of the number of busy servers, and $\sum_{b=1}^{B} N_{b,r} = NB_r$ is the mean number of class (M,r) customers. $R_{b,m}$ and $Q_{b,m}$ are auxiliary variables.

On top of the server utilization and throughputs, the algorithm yields the mean number of class (M,r_o) customers for all r_o .

In order to obtain the mean number of class (m,r) customers for some other m, it is necessary to change the numbering of groups, and perform the algorithm after each re-numbering. (A tree permutation of the groups can avoid performing the whole algorithm).

If mean queue lengths are required for groups only, then a simpler version of the algorithm, that requires less storage, can be used.

The recursions of proposition 2 result in the following proposition :

Proposition 3 (application of the basic recursions)

(i) for $1 \leq b \leq B-1$:

$$N_{b,r}(\underline{K}) = \frac{\lambda_r(\underline{K}) q_{M,r}}{\mu} NA_b(\underline{K}-\underline{e}_r) + q_{M,r} \frac{\lambda_r(\underline{K})}{\mu} (R_{b,M}+Q_{b-1,M})(\underline{K}-\underline{e}_r)$$

(ii) $$NB_r(\underline{K}) = \frac{1}{B}\Big\{ \sum_{b=1}^{B-1} (B-b) N_{b,r}(\underline{K}) + \frac{1}{\mu} \sum_{r'=1}^{R} \lambda_{r'}(\underline{K}) NB_r(\underline{K}-\underline{e}_{r'}) + \frac{q_{M,r}\ \lambda_r(\underline{K})}{\mu} \Big\}$$

(iii) for $1 \leq b \leq B-1$ and $b \leq m \leq M$:

$$R_{b,m}(\underline{K}) = \sum_{r=1}^{R} \frac{\lambda_r(\underline{K}) q_{m,r}}{\mu} (R_{b,m}(\underline{K}-\underline{e}_r)+Q_{b-1,m}(\underline{K}-\underline{e}_r))$$

(iv) for $1 \leq b \leq B-1$ and $b+1 \leq m \leq M+1$

$$Q_{b,m}(\underline{K}) = Q_{b,m-1}(\underline{K}) + R_{b,m-1}(\underline{K})$$

Proof :

The proposition is a direct application of proposition (2) and of the two following relations ([RK 75], [RL 80]) :

$$\begin{cases} P_\ell(\underline{x}_\ell|\underline{K}) = f_\ell(\underline{x}_\ell)\ G^{[\ell]}(\underline{K}-\kappa(\underline{x}_\ell))/G(\underline{K}) \\ \lambda_r(\underline{K}) = \theta_r\ G(\underline{K}-\underline{e}_r)/G(\underline{K}) \end{cases}$$

Note that formula (i) is a generalization of the basic MVA recursion of Reiser and Lavenberg.

Two additionnal relations :

Proposition 4

$$P_B(\underline{K}) = \frac{1}{B}\left(\frac{1}{\mu}\sum_{r=1}^{R}\lambda_r(\underline{K}) - \sum_{b=1}^{B-1} b\ P_b(\underline{K})\right)$$

Proof :

It is a direct application of Little's formula to the multiple server.

Proposition 5

$$P_o(\underline{K}) = G^{[\ell]}(\underline{K})/G(\underline{K})$$

Proof :

It is a well known relation, directly implied by (4-1).

We are now able to give a brief description of the MULTIBUS algorithm : The algorithm consists in a loop over the population vector $\underline{k}$:

- $N_{b,r}(\underline{k})$ for b in {0...B-1}, and $NA_b(\underline{k})$, $NB_r(\underline{k})$ are computed thanks to prop. 3, (i) and (ii), for all chain r_o . At this step, $\lambda_r(\underline{k})$ is known only up to the multiplicative constant $1/G(\underline{k})$.
- $R_{b,m}$ is obtained by means of prop. 3 (iii), for all b and $m \geqslant b$. $Q_{o,m}(\underline{k}) = P_o(\underline{k})$ is obtained from prop. 5, and $Q_{b,m}$ for $b \geqslant 1$ from prop. 3, (iv). This allows to compute P_b for all $b \leqslant B-1$; P_B results from prop. 4.
- The normalizing constant is then derived from the normalizing of the P_b's $0 \leqslant b \leqslant B$.

Notice

1- The computation of the mean number of class (M,r_0) customers involves the values of all $\lambda_r(K)$, $1 \leq r \leq R$. This is due to the interference of customers belonging to different closed chains at the MSCCC station. Because of this fact, it is necessary to compute (and store) the normalizing constants $G^{[\ell]}(\underline{K}')$ and $G(\underline{K}')$, for $K' \leq K$. Such a problem does not arise if there is only one chain of customers (see section 4-4), or if the different chains of customers do not compete in the same group of classes (for instance if, for every r and m, $q_{m,r} = 0$ except for one m(r) where $q_{m(r),r} = 1$).

2- One can allow μ to depend on the total population $|\underline{K}| = K_1 + \ldots + K_R$ in propositions 3 to 5. This brings no change in the MULTIBUS algorithm, except that μ has to be indexed by $|\underline{K}|$.

We give now the MULTIBUS algorithm in full detail :

The MULTIBUS algorithm, multiple chain case

Input:

B, M, mu :	number of servers, classes and service rate of the MSCCC station
$q_{m,r}$, $1 \leq m \leq M$, $1 \leq r \leq R$	probability that a chain r arriving customer joins class m
$theta_r$, $1 \leq r \leq R$	visit ratio for chain r
$\underline{K}$	population vector;
$G^{[l]}(\underline{k})$, $\underline{k} \leq \underline{K}$;	array of normalizing constants for the complement network;

Output:

$G(\underline{k})$, $\underline{k} \leq \underline{K}$;	array of normalizing constants for the network
$NB_r(\underline{k})$, $\underline{k} \leq \underline{K}$: $1 \leq r \leq R$	array of the mean numbers of class (M,r) customers
P_b, $0 \leq b \leq B$	distribution of the number of busy servers, given the population vector is $\underline{K}$;
U	utilization of the servers;
la_r, $1 \leq r \leq R$	throughput for chain r,given the total population vector is $\underline{K}$;

Local:

Q :	array [0:B,1:M ,-1:K1, ... ,-1:KR] of real;
R :	array [1:B,1:M, -1:K1, ... ,-1:KR] of real;
N :	array [0:B - 1,1:R,-1:K1, ... ,-1:KR] of real;
b,m,k :	integer;
NA_b :	real;

Begin

{initialization}

$G^{[l]}(0) := 1$;

for all r_0 , r in {1...R} , all b in {1...B-1} do $N_{b,r0}(-\underline{e}_r) := 0$;

for all b in {0...B-1} do (for all r in {1,...,R} do

($P_b := 0$, for all m in {1...M} do $Q_{b,m}(-\underline{e}_r) := 0, R_{b,m}(-\underline{e}_r) := 0$));

{main loop}

for k_1 from 0 through K_1, ... , k_R from 0 through K_R do

{computation of $N_{b,r}(\underline{k})$ and NB_r }

for all r in {1...R} do

$la_r := G(\underline{k}-\underline{e}_r)\ theta_r$;

for all b in {1...B-1} do

$NA_b := sum(N_{b,r'}(\underline{k} - \underline{e}_r)$, for r' from 1 through R);

$N_{b,r}(\underline{k}) :=$ *proposition 3, (i)* ;

$NB_r(\underline{k}) :=$ *proposition 3, (ii)* ;

{computation of P_b}

{computation of $R_{b,m}(\underline{k})$ and $Q_{b,m}(\underline{k})$ }
for b from 1 through B-1 , m from b through M do
$R_{b,m}(\underline{k})$:= *proposition 3, (iii)* ;
for m from 1 through M do $Q_{0,m}(\underline{k}) := G^{[l]}(\underline{k})$;
for b from 1 through B-1 do
$Q_{b,b}(\underline{k}) := 0$;
for m from b+1 through M do
$Q_{b,m}(\underline{k}) := Q_{b,m-1}(\underline{k}) + R_{b,m-1}(\underline{k})$;

{computation of P_b}
for b from 1 through B-1 do $P_b := R_{b,M}(\underline{k}) + Q_{b,M}(\underline{k})$;
$P_0 := Q_{0,1}(\underline{k})$;
P_B := *proposition 4* ;

{computation of the normalizing constant}
$G(\underline{k})$:= sum(P_b , for b from 0 through B);

{normalization}
divide P_b , $Q_{b,m}(\underline{k})$, $R_{b,m}(\underline{k})$, $N_{b,r}(\underline{k})$, $NB_r(\underline{k})$ for all b,m,r by $G(\underline{k})$;

{end of main loop}

{computation of results}
for all r in {1...R} do $la_r := theta_r\ G(\underline{K}-\underline{e}_r) / G(\underline{K})$;
U := sum (b P_b , for b from 0 through B)

End.

Storage requirement

The arrays that have to be stored are $G^{[l]}(\underline{k})$, $G(\underline{k})$, $N_{b,r}(\underline{k})$, $NB_r(\underline{k})$ together with the auxiliary variables $R_{b,m}(\underline{k})$ and $Q_{b,m}(\underline{k})$. The total storage requirement to perform once the algorithm is thus of the order of K(MBR+2), when the total population $K = K_1 + \ldots + K_R$ is supposed to be much greater than M and B.

Computational effort

The operation count for the first execution of the algorithm is of the

order of 2KBMR multiplications and 2KBMR additions. (K is the total population in the network). If the algorithm is executed several times to obtain results concerning other groups of classes than group M, then the array of normalizing constants $G(\underline{k})$ need not be computed again, which slightly reduces the operation count. A tree structured procedure should significantly reduce the total operation count. ([T-S 85]).

Simpler version of the algorithm

Suppose only the mean number of group M customers is required, and define :

$$N' = \sum_{b=1}^{B} NA_b$$

then proposition 3, (i) and (ii) can be replaced by :

(i bis) for $1 \leq b \leq B-1$

$$NA_b(\underline{K}) = \frac{1}{\mu} \sum_{r=1}^{R} \lambda_r(\underline{K}) q_{M,r} (NA_b(\underline{K}-\underline{e}_r) + R_{b,M}(\underline{k}-\underline{e}_r) + Q_{b-1,M}(\underline{k}-\underline{e}_r))$$

(ii bis) $$N'(\underline{K}) = \frac{1}{B} \{ \sum_{b=1}^{B-1} (B-b) NA_b(\underline{K}) + \frac{1}{\mu} \sum_{r=1}^{R} \lambda_r(\underline{K}) (N'(\underline{K}-\underline{e}_r) + q_{M,r}) \}$$

which avoids the computation and storage of $N_{b,r}(\underline{k})$ for all r.

4-4 The MULTIBUS algorithm, single chain case

If there is only one chain of customers, then the computation (and storage) of the normalizing constants can be replaced by that of the throughputs. For the complementary network $\mathcal{N}^{[\ell]}$ they are related by ([Rei 81]) :

$$\lambda^{[\ell]}(k) = G^{[\ell]}(k-1)/G^{[\ell]}(k) .$$

The algorithm is then quite similar to the algorithm for the multiple chain case ; in a first step, the value of $\lambda(k)$ is set to 1, then the normalizing condition $\sum_{b=0}^{B} P_b(k) = 1$ yields the exact value of $\lambda(k)$.

The MULTIBUS algorithm, single chain case

Input:
- B, M, mu : number of servers, classes and service rate of the MSCCC station
- q_m, $1 \le m \le M$: probability that an arriving customer joins class m
- K : population of the network;
- $la^{[l]}(k)$, $k \le K$: array of throughputs for the complement network;

Output:
- la(k), $k \le K$: array of throughputs for the network
- NB : mean number of class M customers, given the total population is K;
- P_b, $0 \le b \le B$ distribution of the number of busy servers, given the total population is K;
- U utilization of the servers;

Local:
- Q : array [0:B,1:M] of real;
- R : array [1:B,1:M] of real;
- N : array [0:B - 1] of real;
- b,m,k : integer;

Begin

{Initialization}

NB := 0;
for all b in {1...B-1} do (N_b := 0 , for all m in {1...M} do $Q_{b,m}$:= 0 ,$R_{b,m}$:= 0);
for all m in {1...M} do $Q_{0,m}$:= 1;

{main loop}

for k from 1 through K do

{computation of N_b and NB }

for all b in {1...B-1} do
{proposition 3, (i) }
$N_b := q_m / mu\ (N_b + R_{b,M} + Q_{b-1,M})$;

{proposition 3, (ii) }
NB :=((NB+q_m)/ mu + sum((B-b)N_b, for b from 1 through B-1))/B;

{computation of P_b}

{computation of $R_{b,m}$ and $Q_{b,m}$}
for b from 1 through B-1 , m from b through M do
{proposition 3, (iii) }
$R_{b,m} := q_m / mu\ (R_{b,m} + Q_{b-1,m})$;

for m from 1 through M do $Q_{0,m} := Q_{0,m} / la^{[l]}(k)$;
for b from 1 through B-1 do
$Q_{b,b} := 0$;
for m from b+1 through M do $Q_{b,m} := Q_{b,m-1} + R_{b,m-1}$;

{computation of P_b}
for b from 1 through B-1 do $P_b := R_{b,M} + Q_{b,M}$;
$P_0 := Q_{0,1}$;

{ *proposition 4* }
$P_B := 1/B\ (1/\ mu - sum(\ bP_b\ ,$ for b from 1 through B-1));

{computation of the thoughput}
la(k) := 1 / sum(P_b , for b from 0 through B));

{normalization}
multiply NB and P_b , $Q_{b,m}$, $R_{b,m}$, N_b for all b,m by la(k);

{end of main loop}

{computation of the utilization}

U := la(K) / mu

End.

Complexity

N_b need not be indexed by k, which reduces the storage requirement to ca.2K. The operation count for each execution of the algorithm is of the order of 3KBM multiplications and KBM additions.

5 - THE OPEN CASE

Assume now the network to be open (and ergodic). Then, at equilibrium, all queues behave independently. It is thus sufficient to study the MSCCC station in isolation. In that case, the probability that a given customer in group m belongs to class (m,r_o) is $\theta_{m,r_o} / \Sigma_r \theta_{m,r}$ (where $\theta_{m,r}$ is the mean customers of class (m,r) customer arrival per time unit), and the customers of class (m,r_o), for all r_o, behave identically in the station. Hence we restrict ourselves to the single chain case, and drop subscript r.

A first aggregation formula, that slightly differs from (3-1), was given in [LeB 85] ; it provides the normalizing constant and the generating function, but it involves a complicate summation.

An alternative is to use the basic recursion of section 4-2. Nevertheless, no simple closed form solution can be obtained. Similarly to section 4-3, define :

ν_m : the number of group m customer at time t_o

β : the number of busy servers at time t_o

and

$$P_b \stackrel{d}{=} \mathbb{P}(\beta=b)$$

$$N_{b,m} \stackrel{d}{=} \mathbb{E}(\nu_m \, 1_{(\beta=b)})$$

$$N_m \stackrel{d}{=} \mathbb{E}(\nu_m) = \sum_{b=1}^{B} N_{b,m}$$

where $\mathbb{P}$ [resp. $\mathbb{E}$] is the symbol for the steady-state probability [resp. expectation].

Also note :

G : the normalizing constant ;

$y_m = \rho_m/(1-\rho_m)$;

$\rho = \rho_1 + \ldots + \rho_M$.

Proposition 6

(i) for $0 \leq b \leq B-1$ $\quad P_b = \left(\frac{1}{G}\right) \sum_{1 \leq i_1 < i_2 < \ldots < i_b \leq M} y_{i_1} y_{i_2} \cdots y_{i_b}$

(ii) $P_B = \frac{1}{B} \left(\rho - \sum_{b=0}^{B-1} b \, P_b\right)$

(iii) for $1 \leq b \leq B-1$, $N_b = \frac{\rho_M}{(1-\rho_M)^2} \frac{1}{G} \sum_{1 \leq i_1 < i_2 < \ldots < i_{b-1} \leq M-1} y_{i_1} \ldots y_{i_{b-1}}$

(iv) $N = \frac{1}{B-\rho} (\rho_M + \sum_{b=1}^{B-1} (B-b) N_b)$

(v) $G = \frac{1}{B-\rho} (B + \sum_{b=1}^{B-1} (B-b) \sum_{1 \leq i_1 < i_2 < \ldots < i_b \leq M} y_{i_1} y_{i_2} \ldots y_{i_b})$

Proof :

- Define $R_{b,m} = \mathbb{P}(\nu_m > 0, \nu_{m+1} = \ldots = \nu_n = 0)$, $Q_{b,m} = \mathbb{P}(\nu_m = \ldots = \nu_{M+1} = 0)$;

by proposition 2, (ii) :

$$R_{b,m} = \rho_m (R_{b,m} + Q_{b-1,m})$$

thus

(5-1) $$R_{b,m} = y_m \, Q_{b-1,m}$$

The recursion of proposition 2, (iii) now becomes :

$$Q_{b,m} = Q_{b,m-1} + y_{m-1} \, Q_{b-1,m-1}$$

which proves that

$$Q_{b,m} = Q_{o,o} \sum_{1 \leq i_1 < i_2 < \ldots < i_b \leq m-1} y_{i_1} y_{i_2} \ldots y_{i_b} .$$

Now $P_b = Q_{b,M} + R_{b,M} = Q_{b,M+1}$, which proves (i). Formula (ii) is a consequence of Little's formula.

- By proposition 2, (i) :

$$N_b = \rho_M N_b + \rho_M (R_{b,M} + Q_{b-1,M})$$

applying (5-1) :

$$N_b (1-\rho_M) = \rho_M (y_M + 1) \, Q_{b-1,M}$$

which proves (iii).

- Proposition 2, (iv) yields :

$$\sum_{b=1}^{B} b N_b = \rho N + \rho_M$$

which in turn implies (iv).

- Formula (v) is a direct application of (i) and (ii).

Proposition 6 yields the following algorithm :

The MULTIBUS algorithm for the MSCCC station in isolation, single chain case

{computation of G}

for all m in {1...M} do $Q_{o,m} := 1$;
for b from 1 through B-1 do (for m from b+1 through M+1 do
$Q_{b,m} := Q_{b,m-1} + y_{m-1} Q_{b-1,m-1}$)

$G := (B + \text{sum}((B-b)\, Q_{b,M+1},\ \text{for b from 0 through B-1}))/(B-\rho)$

{computation of P_b}

for b from 0 through B-1 do $P_b := Q_{b,M+1}/G$;

$P_B := [\rho - \text{sum}(b\, P_b,\ \text{for b from 1 through B-1})]/B$;

{computation of N}

for b from 1 through B-1 do $N_b := Q_{b-1,M}\, y_M/(1-\rho_M)/G$;

$N := (\rho_M + \text{sum}((B-b)N_b,\ \text{for b from 1 through B-1}))/(B-\rho)$.

6 - A NUMERICAL EXAMPLE

Consider the multiprocessor system of section 2 (fig. 2 and 3). In this section, we give numerical results for the single chain case. The parameters of the example are given below :

* Number of processors : $P = 64$
* Total number of buses $B = 8$, dispatched in G multiple buses with $G \in \{1,2,4,8\}$. Hence each multiple bus consists in $8/G$ buses.
* Number of memory modules : 64. Each multiple bus gives access to $64/G$ memory modules.
* Each processor selects the first memory module with probability $\alpha \in \{\frac{1}{64}, \frac{1}{16}, \frac{1}{4}, \frac{3}{4}\}$ and the rest with uniform probability distribution. (This imposes the values of the routing probabilities from the PROCESSOR station to the BUS stations).
* Mean sojown time at the PROCESSOR station : λ .
* Mean service time at each bus : μ .

The program was written in FORTRAN, run under VAX/VMS. Figures 6 and 7 plot the processing power (PRCPOW) versus μ/λ. PRCPOW is defined to be the mean number of active processors in the PROCESSOR station. Figure 6 shows the influence of α when the number of multiple buses in $G = 4$. (Curves for other values of G are quite similar). In figure 7, α is fixed to 1/64 (uniform memory references), and G varies in $\{1,2,4,8\}$.

Heavy load behaviour

By Little's formula, the throughput THRPUT at the PROCESSOR station is related to PRCPOW by :

$$PRCPOW = THRPUT / \lambda$$

In case of heavy load (small values of μ/λ), there is a bottleneck at one or several buses, and the throughput (and hence PRCPOW) are proportional to μ. This yields the linear parts of the curves in fig. 6 and 7 . In the case of uniform memory references, all buses are saturated, and then :

$$THPPUT \simeq 8\ \mu$$

$$PRCPOW \simeq 8\ \frac{\mu}{\lambda}$$

In the case of extremely unbalanced memory references ($\alpha = 3/4$), only one bus is saturated ; then :

$$\text{THRPUT} \simeq \mu + \mu \times \frac{\frac{1}{4}}{\frac{3}{4}} = \frac{4}{3}\,\mu$$

$$\text{PRCPOW} \simeq \frac{4}{3}\,\frac{\mu}{\lambda} \quad .$$

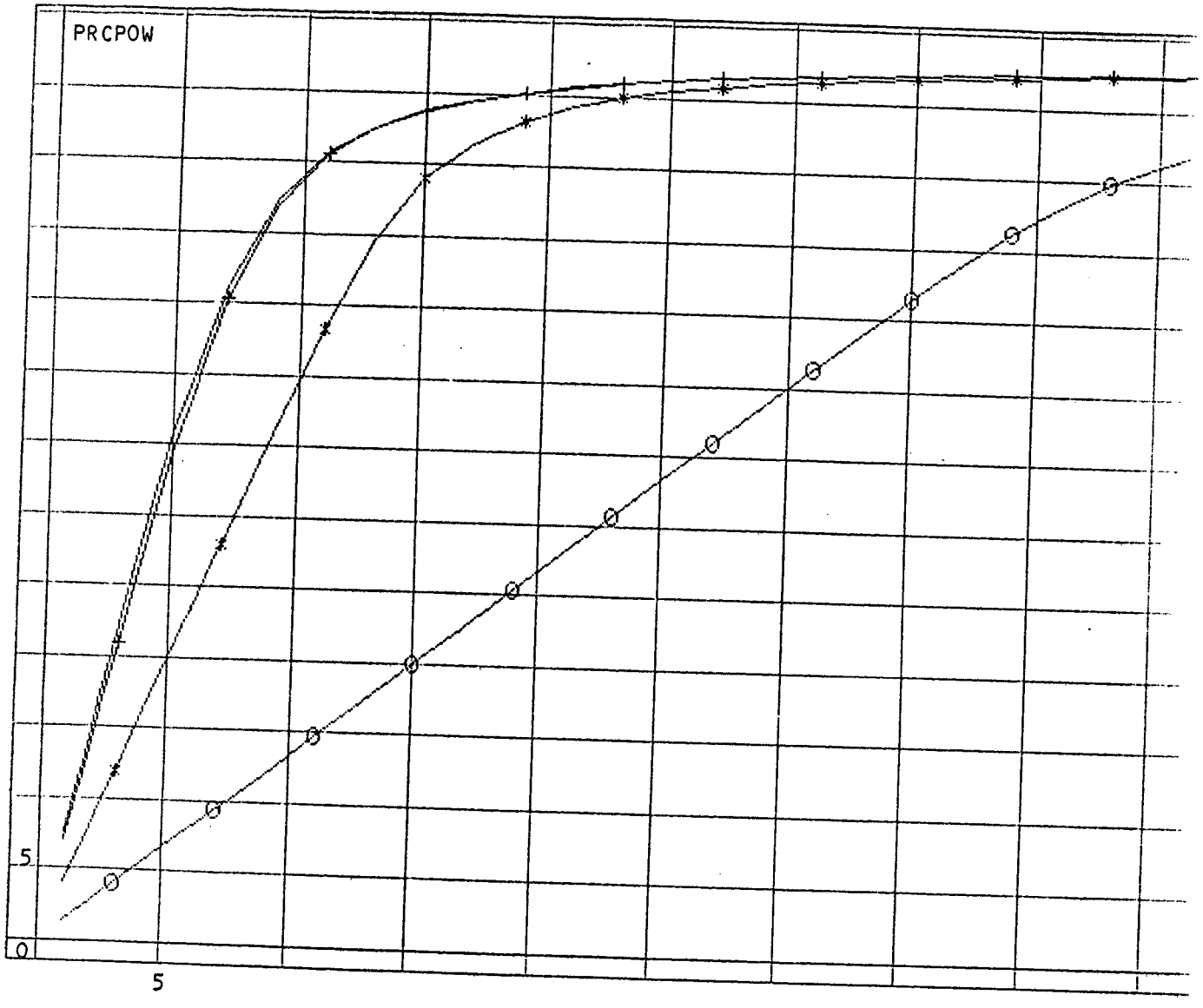


Figure 6 : G = 4, influence of α

$\alpha = 1/64$ $\alpha = 1/4$

$\alpha = 1/16$ $\alpha = 3/4$

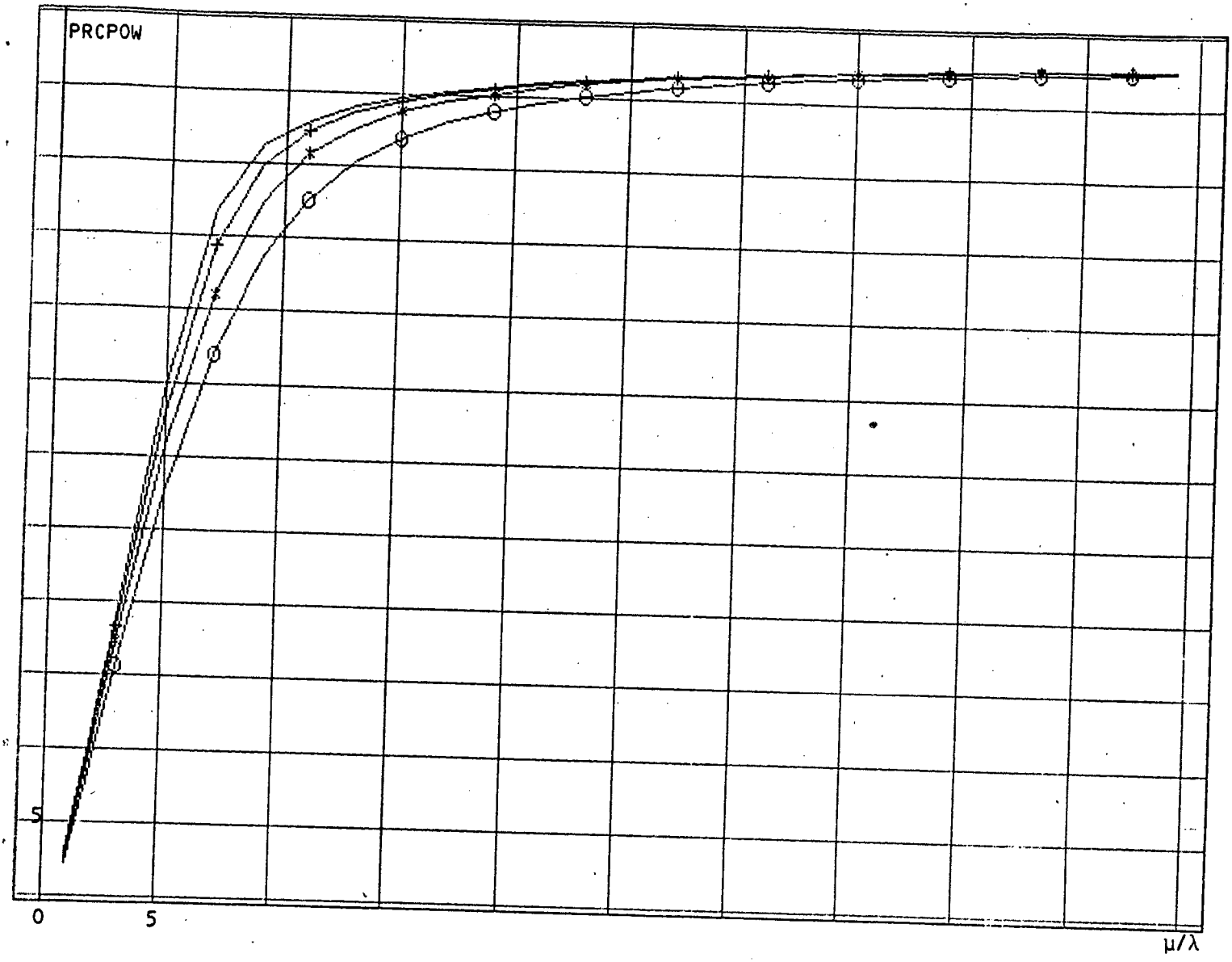


Figure 7 : uniform memory references,
influence of G, number of multiple buses

——— G = 1 —*—*— G = 4

—+—+— G = 2 —0—0— G = 8

APPENDIX

MISCELLANEOUS ABOUT LOCAL BALANCE, AGGREGATION AND EQUIVALENT SERVICE RATES

The appendix checks the validity of some aggregation formulas for product-form networks. It is skown that, under a compatibility assumption (A-5) the equivalent service rates of the station in isolation and in the network are equal ; different notions related to local balance are examined in detail.

Notation (appendix)

e_s : a detailed (or micro-) state of station s ;

$\underline{e}$: a detailed state of the network ;

a_s : an aggregate state of station s ;

$\underline{a}$: an aggregate state of the network ;

n_s : an aggregate state of station s, defined by : $n_s(k)$ = number of class k customers in station s ;

$\underline{n}$: the corresponding aggregate state of the network ;

$f_s(e_s)$: the associate function of station s, for detailed states

$F_s(a_s)$: the associate function of station s, for aggregate states

$$(F_s(a_s) = \sum_{e_s \in a_s} f_s(e_s))$$

$p(\underline{e})$: the steady-state probability of the detailed state e of the network ;

$P(\underline{a}), P(\underline{n})$: the steady-state probability of the aggregate state $\underline{a}$ (or $\underline{n}$) of the network ;

$\mu^*_{k,s}(a_s)$, $\mu^*_{k,s}(n_s)$: the equivalent service rate ;

q_k : the probability that an arriving customer joins the station with class k (station in isolation) ;

$\theta_{k,s}$: the mean number of visits to station s by class k customers .

What is usually called local balance

Consider a network of queues, with numbers 1 to S ; the network is assumed to be described by an ergodic markovian process ; for every station s and every state e of the network, define :

(A-1) $\tau_k(\overset{\uparrow}{e},\overset{\downarrow}{s}) \overset{d}{=}$ the exit rate out of state e resulting from a class k arrival at station s

(A-2) $\tau_k(\overset{\downarrow}{e},\overset{\uparrow}{s}) \overset{d}{=}$ the entrance rate into state e resulting from a class k departure from station s.

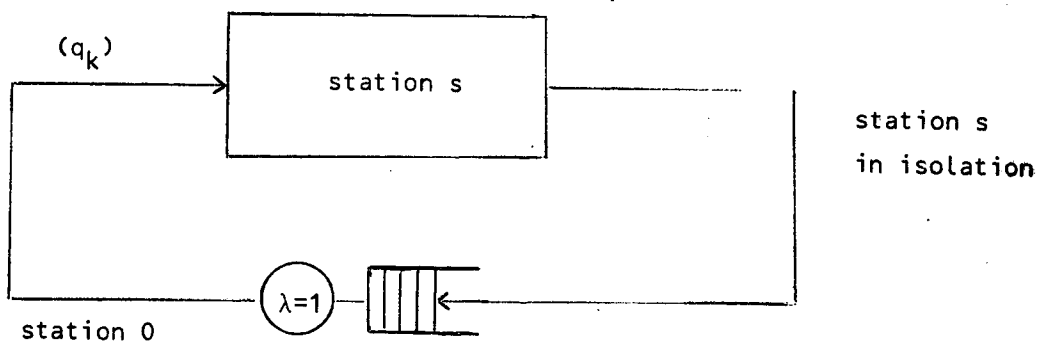
The local balance equation at station s is usually defined by :

(A-3) $\forall k, \forall e \quad \tau_k(\overset{\uparrow}{e},\overset{\downarrow}{s}) = {}_k(\overset{\downarrow}{e},\overset{\uparrow}{s})$

A quasi-reversible queue is a queue at which local balance holds.

Local balance in isolation

Connect station s to an auxiliary one-class exponential station (number 0) with FIFO discipline and λ=1 service rate. The loop obtained is called : "station s in isolation". In the loop, a customer leaving station 0 joins station s, in class k, with probability q_k : the total number of customers is N. A state of the loop is a state of station s .



Station s is said to be locally balanced in isolation if local balance holds for this loop, for every feasible values of the q_k's and of N. This

implies the existence of a function f, independent of both q_k and N, such that $p_N(e) = c_N\, g(e)\, q_1^{n_1} \ldots q_K^{n_K}$, where p_N is the steady-state probability of the loop with N customers, c_N is a normalizing constant, and n_k is the number of class k customers when station s is in state e.

Local balance in product-form networks

Assume network N to be made of stations that are locally balanced in isolation ; the routing policy is probabilistic, with fixed independant Bernoulli trials. Then it is known that the steady-state probability P of the network has product-form, ([Kel],[Pel]...). This remains true under more general routing policies ([Pel],[LeN],[Pittel],...). Yet, local balance does not generally hold at every station of the network. Rather, equations of the following type hold at every station s, for every state e of the network :

$$(A\text{-}4)\quad \forall e \qquad \sum_k \tau(\overset{\uparrow}{e},\overset{\uparrow}{s}) = \sum_k \tau(\overset{\downarrow}{e},\overset{\downarrow}{s})$$

These equations can be called "inverted local balance". Note that inverted local balance generally does not hold by class. ([Pel], [LeB]).

Compatible aggregations

Now consider aggregate states at every station ; note e_s a micro-state of station s, and a_s an aggregate state. When two micro-states e_s and e'_s result in the same aggregate state, this is noted $e_s \equiv e_{s'}$; "$e_s \in a_s$" means that micro-state e_s is part of the aggregate state a_s .

Let E [resp. E_s] be the set of all feasible micro-state of the network [resp. station s] , the set of aggregations at the S stations is said to be compatible with the network iff :

$$(A\text{-}5)\quad \forall e = (e_1,\ldots,e_S) \in E,\ \forall e' = (e'_1,\ldots,e'_S) \in E_1 \times \ldots \times E_S \text{ , } \underline{\text{if}}\ e_s \equiv e'_s$$
$$\text{for all s } \underline{\text{then}}\ e' \in E$$

Roughly speaking, the set of aggregations is compatible with the network if no unfeasible state is introduced. Consider the example of fig 3 : there are two FIFO stations, and two classes of customers. A micro-state of station 1 or 2 is a sequence of classes, whereas an aggregate state is defined by the numbers

of customers in each class. The aggregation is not compatible, since the order of the customers in the network remains unchanged. A higher level aggregation, with $a_s \stackrel{d}{=}$ the total number of customers in station s would be compatible.

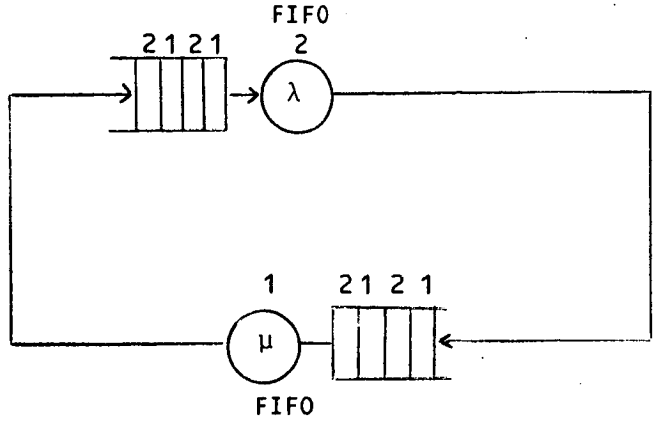


Figure 3

If aggregate state a_s is defined by the number of customers in every class, then a sufficient condition for condition A-5 to be fulfilled is that there exists a PS or IS station visited by every class of customer.

The compatibility condition allows us to consider the cartesian product of aggregate states $\underline{a} = a_1 \times \ldots \times a_S$ as an aggregate state of the network, for which the steady-state probability is given by :

$$(A\text{-}6) \quad P(a) = K\, F_1(a_1) \times \ldots \times F_S(a_S)$$

where $F_s(a_s) = \sum_{e_s \in a_s} f_s(e_s)$ is the result of the aggregation process at station s in isolation.

In other words, if aggregate states are considered at every station, then the associate function of every station can be computed at the aggregate level in isolation, provided the set of aggregations is compatible with the network.

Equivalent service rates and local balance at aggregate level

For a given aggregate state $\underline{a}$ of the network, the equivalent service rate for class k at station s is defined by :

$$(A\text{-}7)\quad \mu^*_{s,k}(\underline{a}) = \sum_{\underline{e}\in\underline{a}} (p(\underline{e})d_{s,k}(e_s)) \Big/ \sum_{\underline{e}\in\underline{a}} p(\underline{e})$$

where $d_{s,c}(e_s)$ is the departure rate for class k, station s, assumed to depend on e_s only. In the networks considered in the appendix, and if the aggregations are compatible with the network, then the equivalent service rate computed in the network is equal to the equivalent service rate computed in isolation. In particular, it depends on a_s only ; from now on it is noted $\mu^*_{s,k}(a_s)$.

Summing local balance in isolation easily results in local balance at the aggregate level ; an interesting case is when aggregate state $a_s = n_s$ is defined by the number n_k of class k customer, for all k ; in that case, local balance in isolation is

$$(A\text{-}8)\quad F_s(n_s)\, q_k = F_s(k^+(n_s))\, \mu^*_{s,k}(k^+(n_s))$$

where $k^+(n_s)$ is the only aggregate state which after a departure yields state n_s, and for all a_s such that $k^+(n_s)$ is feasible. Taking $k^+(n_s)$ as a variable shows that, for all class k :

$$(A\text{-}9)\quad \tau_k(\overset{\uparrow}{n}_s,\overset{\uparrow}{s}) = \tau_k(\overset{\downarrow}{n}_s,\overset{\downarrow}{s})$$

(with the same notation τ_k at aggregate level as at detailed level).

That is, inverted local balance holds by class in isolation. This still holds in the network, as is proven below :

For all state n of the network, the entrance rate resulting from a class k arrival at station 1 is :

$$\tau_k(\overset{\downarrow}{\underline{n}},\overset{\downarrow}{1}) = \sum_{k',s} P^{\{1,s\}}(\underline{n})F_1(k^-(n_1))F_s(k'^+(n_s))q_{s,k',1,k}\,\mu^*_{s,k'}(k'^+(n_s))$$

where $P^A(\underline{n}) = P(\underline{n}) \Big/ \prod_{s\in A} F_s(n_s)$;

By A-8 :

$$\tau_k(\overset{\downarrow}{\underline{n}},\overset{\downarrow}{1}) = \sum_{s,k} P^{\{1,s\}}(\underline{n})F_1(k^-(n_1))F_s(n_s)\theta_{s,k'}\, q_{s,k',1,k}$$
$$= P^{\{1\}}(\underline{n})F_1(k^-(n_1))\theta_{1,k}$$

By A-9 :

$$\tau_k(\underline{\overset{\downarrow}{n}}, \overset{\downarrow}{1}) = P(\underline{n})\ \mu^*_{1,k}(n_1)$$

which is precisely $\tau_k(\underline{\overset{\uparrow}{n}}, \overset{\uparrow}{1})$.

The following table summarizes which relations hold at aggregate and detailed level, assuming the aggregations to be compatible with the network :

	Detailed level	Aggregate level
queue in isolation	$\tau_k(\overset{\uparrow}{e_s}, \overset{\downarrow}{s}) = \tau_k(\overset{\downarrow}{e_s}, \overset{\uparrow}{s})$ (local balance)	$\tau_k(\overset{\uparrow}{n_s}, \overset{\downarrow}{s}) = \tau_k(\overset{\uparrow}{n_s}, \overset{\downarrow}{s})$ $\tau_k(\overset{\uparrow}{n_s}, \overset{\uparrow}{s}) = \tau_k(\overset{\downarrow}{n_s}, \overset{\downarrow}{s})$
queue in the network	$\sum_k \tau_k(\overset{\uparrow}{e}, \overset{\uparrow}{s}) = \sum_k \tau_k(\overset{\downarrow}{e}, \overset{\downarrow}{s})$	$\tau_k(\overset{\uparrow}{n}, \overset{\uparrow}{s}) = \tau_k(\overset{\downarrow}{n}, \overset{\downarrow}{s})$ (inverted local balance)

The omitted relations generally do not hold.

REFERENCES

[B-C-M-P 75] BASKETT, CHANDY, MUNTZ and PALACIOS, "Open, closed and mixed networks of queues with different classes of customers", J. of ACM, vol. 22, n° 2, 248-260, 1975.

[B-B 80] BRUELL and BALBO, "Computational algorithms for closed queueing networks", P.J. Denning Ed., Operating and Programming Systems Series (Elsevier, North Holland, NY, 1980).

[Bhy 84] BHUYAN, "A combinatorial analysis of multiple bus multiprocessors", in Proc. IEEE 1984 Int Conf Parallel Processing, Aug 1984.

[C-H-T 77] CHANDY, HOWARD and TOWSLEY, "Product form and local balance in queueing networks", Journal of the ACM, vol. 24, n° 2, 1977.

[CMB 86] CHIOLA, MARSAN, BALBO, "Product form solutions for the performance analysis of multiple bus multiprocessor systems with non uniform memory references", Research Report, Universita di Torino, Italy.

[Fen 81] FENG, "A survey of interconnection networks", IEEE comput, vol. 14 dec. 81.

[G-P 82] GELENBE et PUJOLLE, "Introduction aux réseaux de files d'attente", Eyrolles, 1982.

[G-A 84] GOYAL and AGRAWALA, "Performance analysis of future shared storage systems", IBM J. Res Develop, vol. 28, n° 1, Jan 1984.

[I-Ö 84] IRANI and ÖNYÜKSEL, "A closed form solution for the performance analysis of multiple bus multiprocessor systems", IEEE trans. on Comp., vol. 33, n° 11, Nov. 1984.

[Jac 63] JACKSON, "Jobshop-like queue system", Management Sci., vol. 10, 131-142, 1963.

[kel 79] KELLY, "Reversibility and stochastic networks", J. Wiley, 1979.

[Lan 82] LANG, VALERO and ALEGRE, "Bandwith of crossbar and multiple bus connections for multiprocessors", IEEE Comput, vol. C31, Dec. 1982.

[LeB 86] LE BOUDEC, "A BCMP extension to multiserver stations with concurrent classes of customers", to appear in Proc. Performance '86, Raleigh, May 1986.

[LeN 80] LE NY, "Etude analytique de réseaux de files d'attente multiclasses à routages variables", RAIRO, vol. 14, n° 4, 331-347, 1980.

[M-B-C-D 84] MARSAN, BALBO, CHIOLA, DONATELLI, "On the product form solution of a class of multiple bus multiprocessors system models", Proc. of the int. workshop on modeling and performance evaluation of parallel systems, Grenoble 1984.

[M-G 82] MARSAN and GERLA, "Markov models for multiple bus multiprocessors", IEEE Comput., vol. C31, March 1982.

[M-H-B-W 84] MUDGE, HAYES, BUZZART, WINSOR, "Analysis of multiple bus interconnection networks", Proc. IEEE 1984, Parallel Processing, Aug. 1984.

[Pel 79] PELLAUMAIL, "Formule du produit et décomposition de réseaux de files d'attente", Ann. Inst. Henri Poincaré, vol. 15, n° 3, 261-286, 1979.

[R-L 80] REISER and LAVENBERG, "Mean value analysis of closed multichain networks", J. ACM, 27(2), 1980.

[Rei 81] REISER, "Mean value analysis and convolution methods for queue dependant servers in closed queueing networks", Performance Evaluation 1(1), 1981.

[T-S 85] TUCCI, SAUER, "The tree MVA algorithm", Performance Evaluation 5 (1985).

PI 290 — **A new matrix multiplication systolic array**
Patrice Quinton, Brigitte Joinnault, Pierrick Gachet – 12 pages ; Avril 86.

PI 291 — **Une introduction à quelques techniques du contrôle distribué à travers un exemple**
Noël Plouzeau, Michel Raynal, Jean-Pierre Verjus – 22 pages ; Avril 86.

PI 292 — **Distributed Synchronization of Parallel Programs : why and how ?**
Patrice Quinton, Jean-Pierre Verjus – 16 pages ; Avril 86.

PI 293 — **Sur l'utilisation des séquences multi – images en robotique**
Lionel Marcé – 16 pages ; Avril 86.

PI 294 — **Stabilité de l'analyse factorielle des correspondances multiples en cas de données manquantes et de modalités à faibles effectifs**
Habib Bénali – 16 pages ; Avril 86.

PI 295 — **The problem of diagnosis with respect to physical parameters for changes in structures**
Georges V. Moustakides – 14 pages ; Avril 86.

PI 296 — **An overview of local environment sensing in robotics applications**
Bernard Espiau – 38 pages ; Mai 86.

PI 297 — **Context – dependent determiners in logic programming : semantic representations and properties**
Patrick Saint – Dizier – 54 pages ; Mai 86.

PI 298 — **Real – Time, Synchronous, Data – Flow Programming : the Language SIGNAL and its Mathematical Semantics**
Paul Le Guernic, Albert Benveniste – 112 pages ; Juin 86.

PI 299 — **The multibus algorithm**
Jean – Yves Le Boudec – 42 pages ; Juin 86.

PI 300 — **Safe implementation equivalence for asynchronous nondeterministic processes**
Boubakar Gamatié – 28 pages ; Juin 86.

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique